

Tekla Structures

Tekla IFC Import to Revit - Client Documentation

Contents	Tekla Structures IFC Import to Revit Structures 2013. Workflow description and background documentation	
Author	Clive Robinson et al	
Date	28 September 2012	
Change History		
Date	Changed by	Issue
16.10.2012	DALA	Removed 18.0 references

TABLE OF CONTENTS

1.	Introduction	3
2.	Steps to Export from Tekla Structures.....	3
3.	Steps to Import to Revit.....	4
4.	Installation workflow	4
5.	Creating the transfer files in Tekla Structures.....	5
6.	Tekla IFC TO Revit Importer Workflow.....	7
7.	Importing Objects.....	9
8.	Object Mapping.....	10
9.	Additional Object Data Mapping	11
10.	Additional Information	12
11.	Known issues.....	Error! Bookmark not defined.
12.	Revit Family mapping.....	13
13.	Advanced user considerations.....	14

DIAGRAMS

Figure 1 – Export to Softwares toolbar	5
Figure 2 – Export to Revit interface	5
Figure 3 – Levels and Location settings	5
Figure 4 – Browser dialogue	6
Figure 5 – Check All and Selected – No issues found.....	6
Figure 6 – Check All and Selected – Issues found	6
Figure 7 – Files completed	6
Figure 8 – Revit interface	7
Figure 9 – Load the Import from Tekla Add-Ins.....	7
Figure 10 – Basic import form	8
Figure 11 – Form with complete options.....	8
Figure 12 – Transfer complete	9

1. Introduction

The standard IFC import utility in the Autodesk Revit Structure 2013 application has some shortcomings, such as all imported elements are created as in place, face-based families based on the generic model family category. In place families are BREP solids that are unique to the particular project. In addition, each time a new IFC file is imported Revit creates a new project thus creating difficulties with round-tripping and differencing models between Tekla Structures 18 and Autodesk Revit Structures 2012 (Revit). All trademarks etc. are acknowledged.

The scope of this development includes typical design level objects such as slabs, floors, foundation elements, beams, columns, walls and platework. This integration tries to achieve a more native transfer to Revit for Level of Detail (LOD) 300 "design models". It does not include bolts and rebar, however, these objects can be exported with the traditional IFC export and opened and saved as a separate Revit model which is linked in separately. These improvements also include classification of levels, project datum offset (x, y, z, rotation), system family definitions and phases.

Tekla IFC Import to Revit Add-In has been developed to improve the import of Tekla Structures IFC files by using the Revit API to create native objects from the data supplied by the Tekla IFC file. The Tekla IFC Import to Revit Add-In is available through the 'Export to Softwares' function and details will be provided on the following pages.

IFC (Industry Foundation Classes) is the buildingSMART standard, which have been developed to define model-based construction objects and activities. Classes are defined to describe a range of object variables that have common characteristics and the standard forms an open communication platform operating across the design and construction sectors.

The principal advantage of the IFC is that the format supports multi-material objects at all stages of the building and construction processes, resulting in the transfer of rich building information. The latest version of the IFC format is 2x3 and all the major software vendors are just receiving their formal accreditations.

For further information in connection with the buildingSMART organization or the IFC standard see their website - <http://www.buildingsmart.org/>

Application Programming Interface (API) technologies are based upon the Microsoft® .NET programming platform and as a very simple summary allows applications to directly communicate with each other.

2. Steps to Export from Tekla Structures

1. Install tools.
2. Launch Tekla Structures version 18
3. Set desired phases in the Phase Manager of Tekla Structures. If Revit model has the same phase name as the object has then the object will be set to that phase. If not then nothing is set and "New Construction" or default phase created will be used.
4. Launch 'Export to Revit' from the 'Export to Softwares' toolbar.
5. Set up levels. Start by entering the lowest level and work up the building
6. Set up project datum offset if you have already moved the Tekla Structures model closer to Tekla origin even though Revit reference model is further away. If you have kept the Tekla Structures model on top of the Revit reference model then there should be no reason to offset.
7. Select objects to be exported by 'user selection' or by 'all'. By selection is best since you may not wish to export certain object types.
8. Select Check model. Note that curved members and complex geometries will be classified as BREPS. They will go into Revit as families with the IFC property set attributes. The user can choose to include them/exclude them through the relevant selection.
9. Export. It creates an ifcZIP file which also includes all the associated files.

3. Steps to Import from Tekla

1. Launch Revit Structure 2013
2. Preparation - Create model based on the users requirements including:
 - a. Units - imperial or metric
 - b. Phases - created phases with same name as in Tekla Structures
 - c. Materials with same name as in Tekla Structures. If material does not exist it will be created with default settings.
3. It is possible to create levels in Revit but if they are created in Tekla Structures it is important to remove levels from the base Revit model before importing.
4. Go to Add-Ins tab and select 'Import from Tekla'.
5. Browse to the required ifcZIP file.
6. Set options for importing 'All' or by 'element type'. Apply Fittings means that objects will have Tekla cuts/fittings/openings applied in Revit. Set the Limits fields if the model is large and you want to import the project with various files.
7. If a large model the import could take several minutes or longer.

4. Installation workflow

The Tekla IFC to Revit Importer is supplied with an MSI installer.

Basic installation procedure:

1. Copy the setup.exe and ImportFromTeklaToRevit_2013_(version)_x64.msi to your local hard drive.
2. Double click the msi to run the installer and follow the basic installation information.
3. When asked for the path to install files it is necessary to supply the path for the Revit Add-Ins file. In a standard Revit installation this will be; C:\ProgramData\Autodesk\Revit\Addins\2013

When the installation has finished the following directory structure will be created:

- Installation of the Revit Add-Ins file;
 - C:\ProgramData\Autodesk\Revit\Addins\2013\ ImportToRevit.addin
- Installation of all program files and dlls;
 - C:\ProgramData\Tekla\Revit-2013-Interoperability\ImportToRevit
- Installation of the Metric and Imperial directories (and Revit families)
 - C:\ProgramData\Tekla\Revit-2013-Interoperability\ImportToRevit\ImperialFamilies
 - C:\ProgramData\Tekla\Revit-2013-Interoperability\ImportToRevit\MetricFamilies
- Family templates and individual directories for all on the fly created families including beams, columns, plates, voids, footings etc;
 - C:\Users\user\AppData\Local\Tekla\Revit-2013-Interoperability\Environment\ImperialFamilies\
 - C:\Users\user\AppData\Local\Tekla\Revit-2013-Interoperability\Environment\MetricFamilies\
- Location of the run error and run reporting logs;

- C:\Users\user\AppData\Local\Tekla\Revit-2013-Interoperability\Environment

5. Creating the transfer files in Tekla Structures

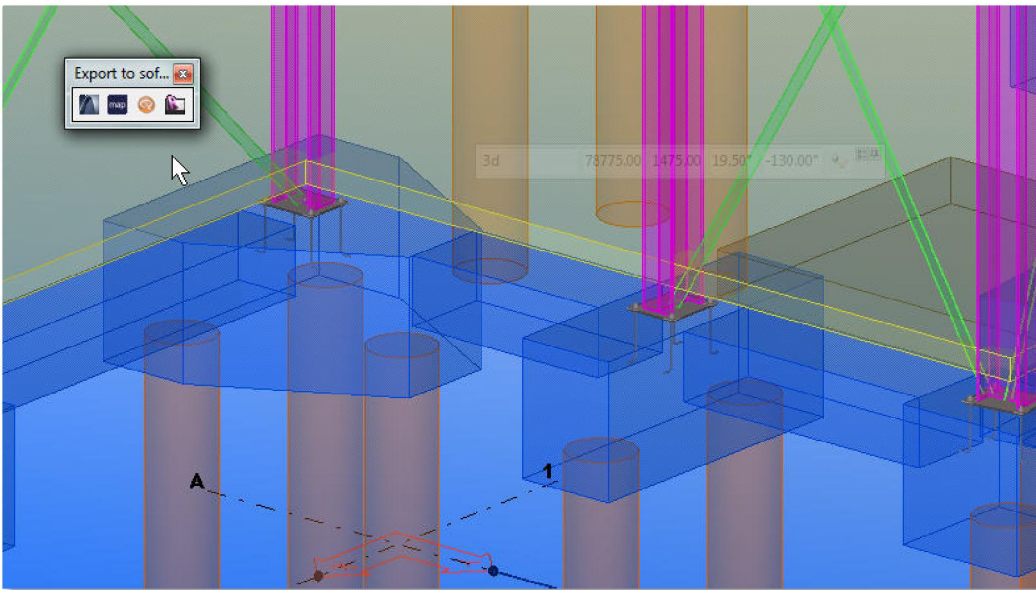


Figure 1 – Export to Softwares toolbar

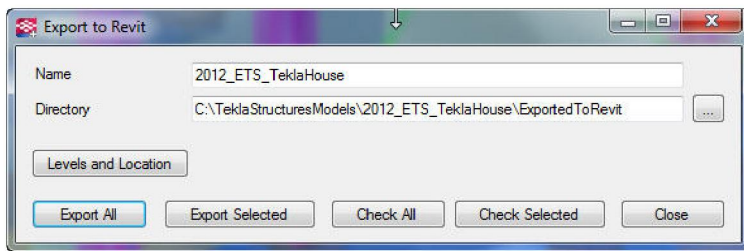


Figure 2 – Export to Revit interface

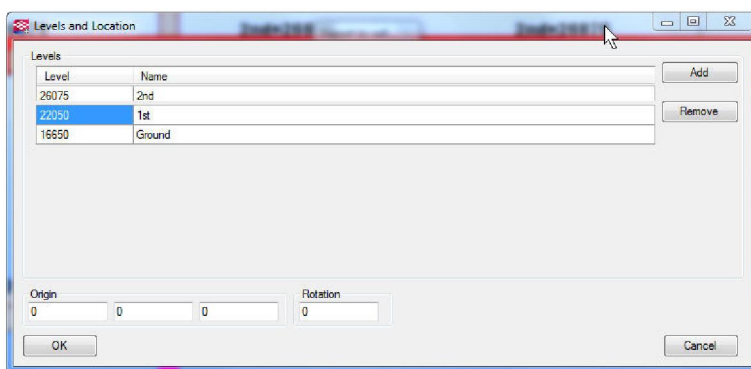


Figure 3 – Levels and Location settings

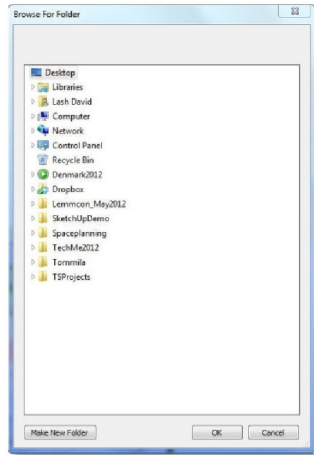


Figure 4 – Browser dialogue



Figure 5 – Check All and Selected – No issues found

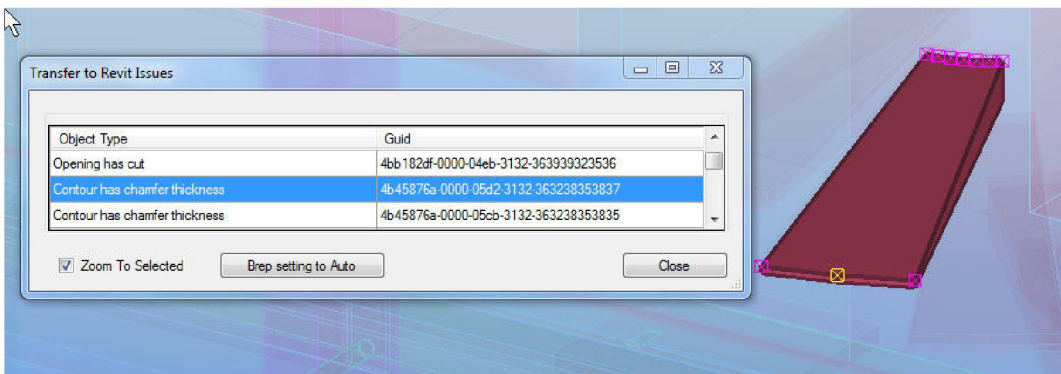


Figure 6 – Check All and Selected – Issues found

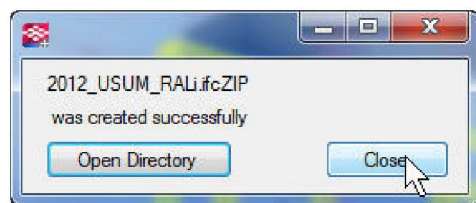


Figure 7 – Files completed

6. Tekla IFC TO Revit Importer Workflow

The following installation is for the Tekla IFC to Revit Importer (after running the installer) open up Revit Structures 2013 and select Add-Ins from the toolbar at the top of the Revit User Interface (UI), the Import From Tekla ribbon button should be visible.

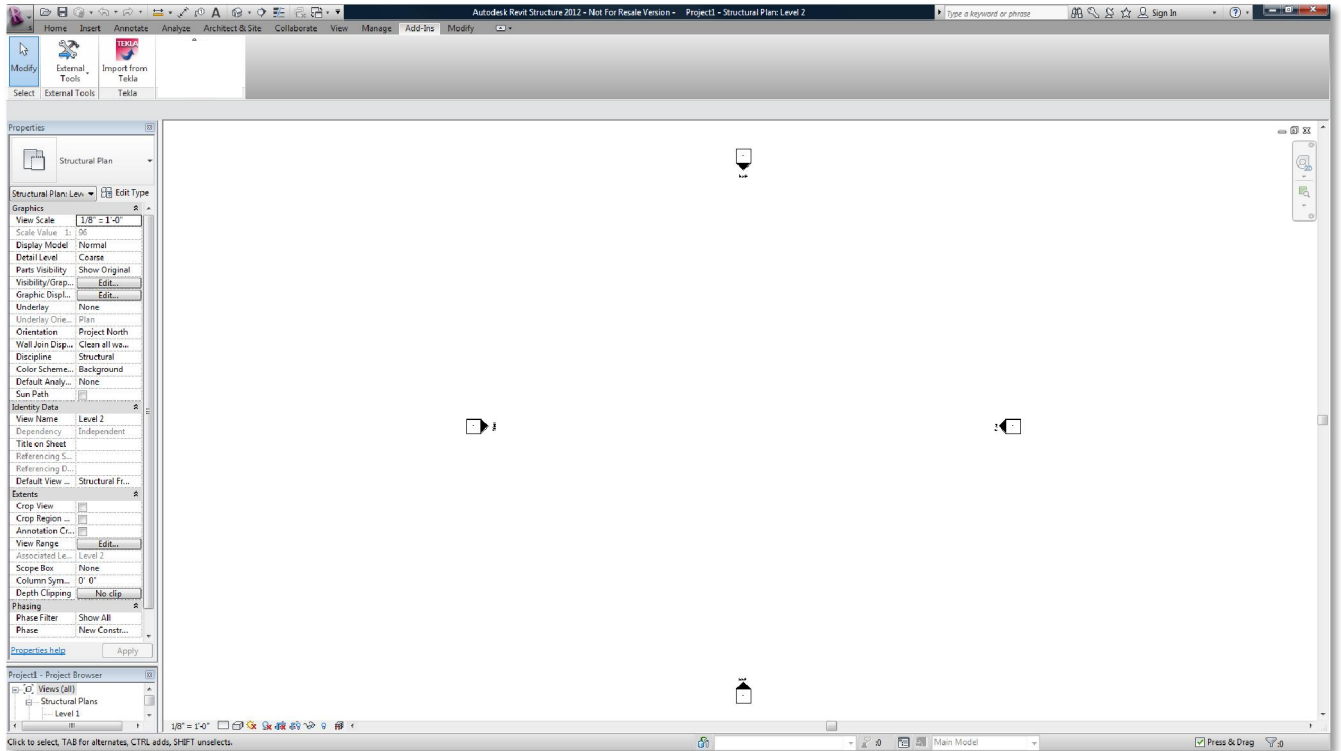


Figure 8 – Revit interface

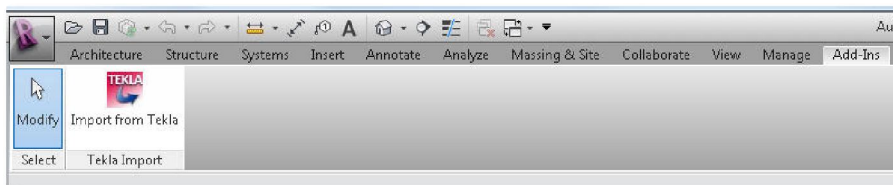


Figure 9 – Load the Import from Tekla Add-Ins

Click the button to start up the Importer UI;

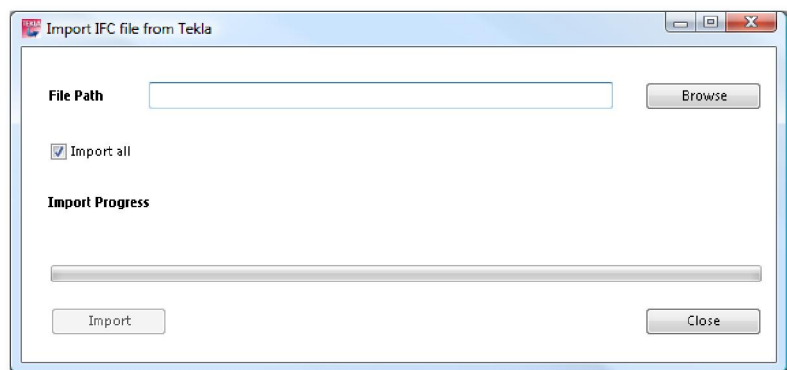


Figure 10 – Basic import form

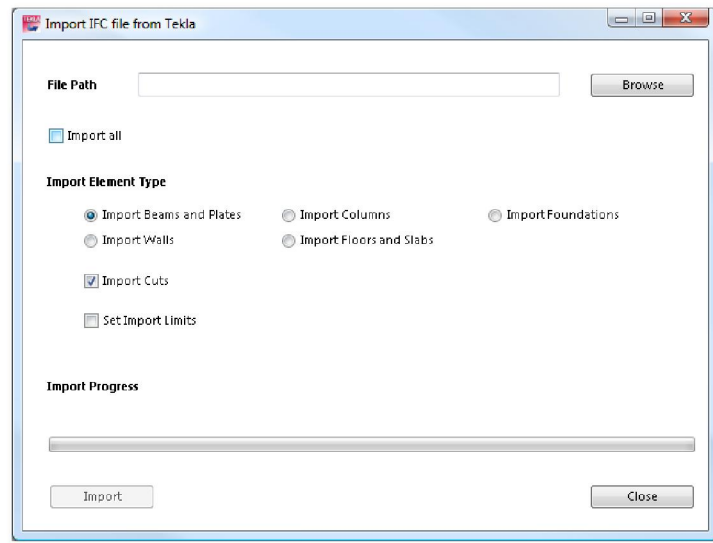


Figure 11 – Form with complete options

The importer UI has the following controls:

- Browse for the required Tekla IFC file (button control, file browse dialog and text box). The importer supports *.ifc and *.ifcZip files. However, the *.ifcZip is the recommended format for file transfer.
- Check box control to either import all elements or, if unchecked, reveal a set of controls to filter for particular elements, as described below.
- Radio button options to filter the imported element types; filter for beams and plates, columns or walls (etc.) only. Default setting is to import all element types.
- Check box control to select whether or not to import cuts. Default is to import cuts.
- Import progress text and progress bar controls. These controls feedback runtime information to the user on the import progress and current situation.
- Import button. Press to start the import.
- Close button. Closes the UI before the import started or after the import has finished. Note - multithreading is not supported in the Revit API, so the user has to wait until the import has completed before closing the UI (or close the Revit process via the Windows Task Manager!).

Once a suitable IFC file has been loaded the user may press the import button and wait for the import to finish. Whilst certain errors and warnings are handled internally by the importer, it is possible that Revit will require some user input to resolve some import issues at runtime. These issues should be resolved as soon as possible otherwise the import will hang, awaiting the decision on how to proceed.

The user will be notified when the import has completed.

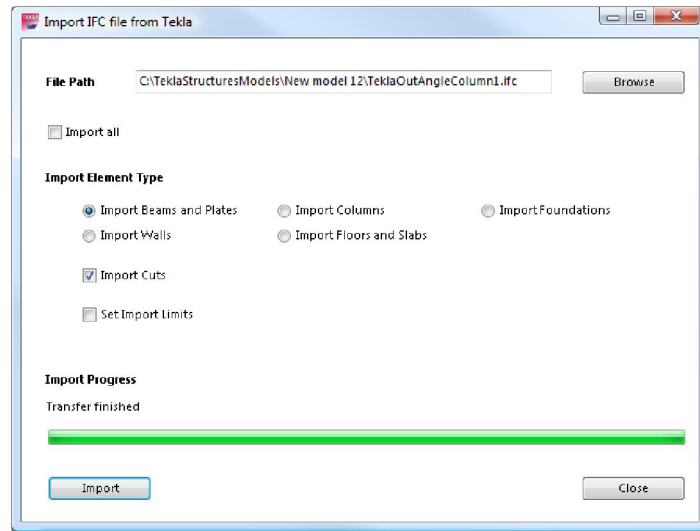


Figure 12 – Transfer complete

Once complete, close the UI and the imported model will be visible in a newly created 3D view in Revit (or the existing 3D view if one is already available).

It may be useful to check the run log in the following location:

C:\ProgramData\Tekla\Revit-2013-Interoperability\ImportToRevit\Environment

The run file will be named depending in the selection of imported elements (from the UI). For example if the Import all element type radio button is selected then the run log file will be named; RunReport_All.txt. If columns only are selected the run log file will be named RunReport_Columns.txt. The log file will be containing a standard header and a time stamp, for example;

```
Tekla Import to Revit run log file
Date and time: 09/20/2012 10:52:46
Warning - no levels file found with this import
```

Any warnings or errors encountered will be logged to the file. Note that this log file is overwritten for each subsequent import.

7. Importing Objects

When the Tekla IFC importer is run the selected IFC file is read by the Tekla 'object reader' and a set of data objects are created. Each IFC building element read in the IFC file is translated to a data object that supplies information on member type, whether parametric, its shape, start and end points, extrusion length etc. Additional information is available for each member that details any associated cuts or voids and attribute data such as member parameters and GUID's.

Import from Tekla supports the import of both parametric and arbitrary defined extruded profile members. BREP objects are imported by using individual DWG files.

Parametric members are created in Revit by mapping to existing Revit parametric families and duplicates are created using the data supplied by the object reader. For non-symmetrical parametric members the analytical wire is placed at the mid depth centre of the section to match the Tekla Structures specification (rather than the centroid of the section as preferred by Revit Structures). Modifications are also made to the analytical wire position to account for the top flange handle point flag in the object converter. Duplicate sections are stored and reused when necessary during the import run. Creating duplicate sections from existing Revit parametric sections and reusing these duplicates is a possibility, and is quicker than creating new parametric families.

Arbitrary defined extruded profiles, supplied by the object reader, are used to create new families which are then loaded into the current project. Depending in the IFC type, either beam or column etc. the appropriate family category template is selected, a profile specified and extruded as a solid. The family is then saved and loaded back into the project so that a family instance can be placed appropriately.

The new extruded families are reused whenever possible, to avoid recreation of the family and speed up the import process.

Voids and cuts are created in a similar fashion to the creation of arbitrary defined extruded profiles. This void is then saved and loaded back into the project and used to create a void by associating the void with the solid to be cut. As the Revit API does not implement the cutting plane tool available in the UI, the cutting plane passed by the object reader has to be converted to an arbitrary profile and extruded. Once a 'cut void' has been created it is applied to the associated solid to create a cut.

Any BREP element in the Tekla model will be exported (when using the Tekla to Revit exporter from Tekla) as an individual AutoCAD dwg format file. Individual families are based on the DWG's that are included in the original exported ifcZIP.

Member attribute or parameter data written by Tekla in the IFC file is transferred to the individual Revit family instances and displayed on the family instance member grid.

8. Object Mapping

For parametric sections; the imported object shape is mapped to the equivalent existing Revit parametric family category as follows:

Beams	-	Structural Framing Family
Columns	-	Structural Column
Pad footings	-	Based on Structural Column family converted to Structural Foundations
Strip Footings	-	Structural Foundations
Pile	-	Structural Foundations
Walls	-	System family - Walls
Slabs (Horizontal)	-	System family - Slabs
Slabs (not horizontal)	-	Converted to arbitrary profile and loaded via the Generic Model family template.

For imported arbitrary sections;

Beams	-	Derived from file based on Structural Framing template
Columns	-	Derived from file based on Structural Column template
Pad / Strip Footings	-	Derived from file based on Structural Foundations template
Pile	-	Derived from file based on Structural Foundations template
Walls	-	Derived from file based on the Generic Model template.
Slabs	-	Derived from file based on the Generic Model template.
Plates	-	Derived from file based on the Generic Model template.

Voids - Derived from file based on Generic Model template.

9. Additional Object Data Mapping

Tekla IFC files exported specifically for use by the Tekla IFC to Revit Importer will contain the following files:

- filename.ifc
- filename _levels.xml
- "guid".dwg as individual brep object

The levels used by this model will be written to the filename _levels.xml file and read by the importer which will create the appropriate levels in the project. The member data available from object reader will allow the member to be assigned to the correct level. If no levels file is provided the importer will warn the user in the run log and use the existing default levels set in Revit.

The object reader will also pass phase data for each member. At present, the Revit API does not allow phases to be created or duplicated so the current rule is that if a member is given a phase name equal to "Phase XXX" in Tekla Structures and an identical Phase exists in Revit, then the phase is set to "Phase XXX" otherwise the phase is set to " New Construction ".

Member materials are passed from Tekla to Revit and the importer searches for an existing material specification that matches. If no match can be found a similar material specification is found and duplicated, renamed and assigned to the member material property. The material cut pattern is also set to the relevant material type, steel or concrete etc.

10. Additional Information

The following provides the user with additional information that may be useful when using the importer:

- The speed at which imports progress can be an issue. Whilst the importer reuses sections whenever possible it appears that modifying existing families or creating new families is a time-consuming process. Every time these operations are necessary the family has to be saved to disk and reloaded into the project. For an IFC file containing several thousand members, with potentially many voids and cuts, this operation may occur hundreds or thousands of times. Whilst in theory the modified family should be able to be reloaded into the project from memory it appears necessary to save the file to disk before the API will accept the changes to the family. In addition, prior to any members being loaded the importer has to potentially create a copy of all parametric families to be used so that these families can be modified with the member parameter data supplied via the IFC file (rather than overwrite the existing default families and force the Revit user to use family symbol instances with dozens of additional parameters etc.). If these families have already been created then the importer will skip the creation of these families but it still has to load all these families into the project, thus some delay is experienced before members are actually loaded.
- When creating cutting plane voids it was found that if the void face intersects with a vertex point on the solid to be cut, the cut action will fail and Revit will request that the void is deleted. If during the solid cutting process the code detects a void face / solid vertex intersection the importer will attempt to apply a small 'nudge' to the void (in the direction of the void extrusion). This void cut will be reattempted three times, increasing the nudge distance by 0.3mm each time. If after three attempts the cut still fails the transaction will be rolled back and the void instance deleted. Any attempt to move the void and the size of the movement and associated solids GUID will be reported to the run log file.
- As the plane cutting tool, available in the Revit UI, is not available in the API it has been necessary to create large voids, derived from the cutting plane, to cut the solid. Prototype code has been developed that derives the profile that defines the cutting plane (supplied by the object reader) intersection with the relevant extruded solid family instance faces. This provides a neat set of points that may be used to define an extruded solid to create a well fitted cutting void. Unfortunately, at present the Revit API only allows solids or voids to be created that have an extrusion axis perpendicular to the cutting or sketch plane provided. As such the use of large voids is still required. Nevertheless, all voids are automatically hidden in all existing views (by the importer) and if the user creates a new view they can easily hide all voids as the voids are grouped into the same category.
- Investigations were undertaken to determine whether arbitrary defined beam profiles could be converted to line based Revit families that have 'handles'. This isn't feasible for the creation of arbitrary extruded profile beams. In order to do this the family needs to contain some general description of the geometry / shape of the particular element (so the shape can be parameterized and glued to reference planes etc). With the arbitrary profiles this shape could be anything and the system doesn't necessarily have any idea of what the shape is, just the profile point data. Of course the IFC parametric shape definitions are a different matter and they can be mapped to the appropriate Revit family type that defines their basic shape. Strip footing may be more suited to this due to the uniformity of their shape (mainly rectangular profiles).
- Particular members (such as wall, slabs etc.) use parameter information and length units to describe the member name. The importer will determine the unit set used either imperial or metric, from the project template and apply the correct unit conversions and units. This technique will be applied to the member parameter data when the object reader is updated to provide each parameter particular units.
- Whilst testing the importer an issue has been noted with the import of dwg files into Revit. When, for example a non-symmetrical section is imported without rotation and a dwg file overlaid, to check the accuracy of the importer, the positions match okay. When a rotation is applied and the IFC file is imported the member will appear correctly rotated about the mid depth centre of the section. The dwg import of the same model shows that the section will be rotated about the centroid of the section.

- Occasionally, Revit will fail to cut a void in a member, although the API will not issue any information about why the cut failed. This is a relatively rare occurrence.
- Occasionally, the Import from Tekla will fail to read a void for a particular element. In addition, naming conventions used may cause some confusion as to whether a particular member has been correctly imported.

11. Revit Family mapping

Family categories will be mapped with default settings. Later there may be user defined mapping available.

IFCObjectsReader: MaterialType - IFCObjectsReader: Material = "CONCRETE/3000"

IFCObjectsReader: Name - IFCObjectsReader: Name = "BEAM" parse as Beam

IFCObjectsReader: ObjectType - IFCObjectsReader: ObjectType = "strip_footing" parse as Strip Footing if reqd.

IFCObjectsReader: ElementType - IFCObjectsReader: ElementType = IfcWallStandardCase

In case of parametric profiles

ElementType (in IFC)	Category	Family	Type
IfcBeam	Structural framing	MaterialType - Name - Profiletype	Profile
IfcBuildingElementPart	Structural framing	MaterialType - Name - Profiletype	Profile/Thickness
IfcBuildingElementProxy	Structural framing	MaterialType - Name - Profiletype	Profile/Thickness
IfcColumn	Structural Columns	MaterialType - Profiletype	Profile
IfcFooting	Structural Foundations	MaterialType - Name - Profiletype	Profile/Thickness
IfcMember	Name (Stairs, Roofs, Ramps) **)	MaterialType - Name - Profiletype	Profile
IfcPile	Structural Foundations	MaterialType - Name - Profiletype	Profile
IfcPlate	Generic Models	MaterialType - Name - Profiletype	Profile
IfcRailing	Railings	MaterialType - Name - Profiletype	Profile
IfcSlab	Floors	MaterialType - Name (Floor)	Thickness
IfcWall/ IfcWallStandardCase	Walls	MaterialType - Name (Basic Wall)	Thickness

In the case of arbitrary profiles

ElementType (in IFC)	Category	Family
IfcBeam	Structural framing	MaterialType - Name - ProfileName
IfcBuildingElementPart	Structural framing	MaterialType - Name - ProfileName
IfcBuildingElementProxy	Structural framing	MaterialType - Name - ProfileName
IfcColumn	Structural Columns	MaterialType - Name - ProfileName
IfcFooting	Structural Foundations	MaterialType - Name - ProfileName
IfcMember	Structural framing	MaterialType - Name - ProfileName
IfcPlate	Generic Models	MaterialType - Name - Thickness (-id)
IfcPile	Structural Foundations	MaterialType - Name - ProfileName

*) should be divided to different kind of objects. No generic possible. Structural Stiffeners/connections would be preferred.

***) vulnerable may not be possible since categories are fixed and name is user given string

Note - Phases are mapping by name. Phases will be read from Phases.xml and no information is available from the IFC file etc.

12. Advanced user considerations

- At present, voids are failing to cut for the creation of say, arbitrary profile square hollow sections. This needs further investigation.
- Large model imports use excessive amounts of random access memory. This appears to be a Revit API issue that needs addressing with Autodesk.
- Load times are slow for models with lots of voids, arbitrary defined profile families and dwg / BREP based families. In order to load a family into the project it first has to be created, saved to disk and then loaded (the procedure of having to save the file to disk rather than simply have the family loaded into memory before loading the family into the project seems to be a known issue with Revit).
- It is recommended that the user, at the very least, adhere to the system requirements specified by Revit in order to successfully load projects.
- The user is advised to check error logs etc from the Tekla Structures export to determine the quality of the export and to view the exported IFC file in a third party viewer such as Solibri Ifc viewer (to ascertain what has actually been exported from the original model).
- The user is advised to check the run log following a Revit import to determine what issues may have arisen during the import process.
- Arb profile elements are not yet line based families.
- Some parametric sections may fail to duplicate in Revit (due to Revit objecting to the parametric data supplied by Tekla). If this occurs an entry will be posted in the run log.
- Slow loading of projects with many dwgs BREP files. Currently each dwg is issued with a unique identifier to be mapped to an element in the ifc file. It would be beneficial to have a one dwg to many ifc entity relationship with the exported files from Tekla Structures if possible.
- It is not possible to duplicate or create phases in Revit via the API so the user either has to set up an appropriate project with the required phases or ensure that the exported model from Tekla Structures uses the existing phases in the Revit project. If phases do not match then Revit will simply assign the default phase to the element.
- Revit will not create system floor families for any floor element that is not absolutely horizontal. If a floor element is exported from Tekla Structures with a slight angle then the IFC importer will use the generic model template to create the element rather than the system floor family.
- A similar approach is taken with wall elements.

-
- Tekla Structures attribute data is transferred to the relevant family instances. In order to create the new parameter templates for each category of family from the IFC data a new set of families is generated for each import that is undertaken. This is the only way to create family instances with sets of parameter data relevant to the information passed for any particular import.